Expert Systems with Applications 38 (2011) 12094–12100

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A random walk algorithm for automatic construction of domain-oriented sentiment lexicon

Songbo Tan *, Qiong Wu

Key Laboratory of Network, Institute of Computing Technology, Chinese Academy of Sciences, China

ARTICLE INFO

Keywords:
Sentiment analysis
Opinion mining
Information retrieval
Data mining

ABSTRACT

In recent years, many studies have been conducted to deal with automatic construction of domain-oriented sentiment lexicon. However, most of the attempts rely on only the relationship between sentiment words, failing to uncover the mutual relationship between the words and the documents, as well as ignoring the useful knowledge of some existed domains (or "old domain"). This paper proposes a random walk algorithm to construct domain-oriented sentiment lexicon by simultaneously utilizing sentiment words and documents from both old domain and target domain (or "new domain"). The approach simulates a random walk on the graphs that reflect four kinds of relationships (the relationship between words, the relationship from words to documents, the relationship between documents, the relationship from documents to words) between documents and words. Experimental results indicate that the proposed algorithm could dramatically improve the performance of automatic construction of domain-oriented sentiment lexicon.

1. Introduction

Sentiment classification (Tan, Cheng, Wang, & Xu, 2009; Tan, Wu, Tang, & Cheng, 2007; Tan & Zhang, 2008; Tang, Tan, & Cheng, 2009; Wu et al., 2009) is the process of identifying the opinion (e.g. negative or positive) of a given document. With the increasing reviewing pages and blogs etc., this field has received considerable attention from researchers in recent years and has many important applications, such as determining critics' opinions about a given product and summarization (e.g. Ku, Liang, & Chen, 2006).

To date, a variety of methods have been developed for sentiment classification. One of the commonly-used methods is to build a sentiment lexicon. However, it is an impossible task to build a general sentiment lexicon that could perform well in every domain. This is because sentiment expression often behaves with strong domain-specific nature. For example, "portable" and "delicate" are used to express positive sentiment in electronics reviews; while these words are hardly used to convey the same sentiment in hotel reviews. Consequently, the domain-oriented sentiment lexicon is considered as the most valuable resource for sentiment classification tasks.

In recent years, two kinds of approaches have been proposed to deal with this problem: one is based on a thesaurus (Esuli & Sebastiani, 2005; Hu & Liu, 2004; Kamps, Marx, Mokken, & Rijke, 2004; Kim & Hovy, 2004); the other one is based on co-occurrence in a corpus (Du, Tan, Cheng, & Yun, 2010; Gamon & Aue, 2005; Hatzivassiloglous & McKeown, 1997; Kanayama & Nasukawa, 2006; Popescu & Etzioni, 2005; Turney, 2002). However, these kinds of approaches rely on only the labeled seed words to construct a domain-oriented sentiment lexicon, failing to uncover the mutual relationship between the words and the documents; besides, these kinds of approaches ignore the labeled lexicon and labeled corpus in one existed domain (or "old domain").

In fact, the polarity of a sentiment word can be determined by the related documents as well as by the related words, and this rule also holds when determining the polarity of a document. This rule is based on the following intuitive observations:

(1) A word strongly linked with other positive (negative) words could be considered as positive (negative); in the same way, a document strongly linked with other positive (negative) documents could be considered as positive (negative).
(2) A word appearing in many positive (negative) documents could be considered as positive (negative); similarly, a document containing many positive (negative) words could be considered as positive (negative).

Furthermore, labeled data in different domains are very imbalanced. In some traditional domains or domains of concern, many labeled data are freely available on the Web, but in other domains, labeled data are scarce and it involves much human labor to manually label reliable data. As a result, the ideal scheme is to utilize

* Corresponding author. Address: Key Laboratory of Network, P.O. Box 2704, Beijing, 100190, PR China. Tel.: +86 10 62600928; fax: +86 10 62600905.
E-mail address: tansongbo@software.ict.ac.cn (S. Tan).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.02.105

labeled data in old domain to assist the construction of the domain-oriented lexicon for new domain.

Inspired by these, we aim to take into account all the four kinds of relationships among words and documents (i.e. the relationship between words, the relationship from words to documents, the relationship between documents, and the relationship from documents to words), from old domain as well as from new domain, when constructing a domain-oriented sentiment lexicon.

In this work, we propose an approach based on random walk to implement the above idea. The proposed approach makes full use of all the relationships among words and documents from both old domain and new domain. The detailed procedure can be divided into four steps: Firstly, four graphs are built to reflect the above relationships respectively. Then, we assign a score for every word to denote its extent to "negative" or "positive". Thirdly, we iteratively calculate the score by simulating a random walk on the graphs. Lastly, the final score is achieved when the algorithm converges, so we can determine the semantic orientation of new-domain words based on these scores.

The rest of this paper is organized as follows. Section 2 surveys related work. In Section 3, we introduce the proposed approach in detail. And then Section 4 presents and discusses the experimental results. Lastly we conclude this paper and discuss future work in Section 5.

2. Related work

So far, some studies have been conducted to deal with the lexicon construction problem. Most of these utilize some paradigm words and word similarity to construct lexicon. According to the manner of obtaining word similarity, these methods could roughly be classified into two categories: the first kind of approaches is based on the thesaurus; the second kind is based on the corpus.

2.1. Thesaurus based approach

Thesaurus based approach utilizes synonyms or glosses of a thesaurus to determine polarities of words.

Kim and Hovy (2004) used the synonym and antonym lists obtained from Wordnet to compute the probability of a word given a sentiment class (i.e. positive or negative). Kamps et al. (2004) made a hypothesis that synonyms have the same polarity. And they linked synonyms provided by a thesaurus to build a lexical network, so the word polarity can be determined by the distance from seed words ("good" and "bad") in the network. Hu and Liu (2004) extended the method in Kamps et al. (2004), and they used not only synonyms but also antonyms to build lexical network. Esuli and Sebastiani (2005) determined the orientation of subjective terms based on the quantitative analysis of the glosses of subjective terms, i.e. the definitions that these terms were given in online dictionaries.

2.2. Corpus based approach

Corpus based approach is based on an idea that sentiment words conveying the same polarity co-occur with each other in corpus. Many studies have been done on this field.

Hatzivassiloglous and McKeown (1997) constructed a lexical network from intra-sentential co-occurrence, and identified the positive or negative semantic orientation of the conjoined adjectives. Turney (2002) determined the semantic orientation of a phrase by the mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor". The mutual information was measured by the number of hits returned by a search engine. Gamon and Aue (2005) extended Turney's approach, and they added one assumption that sentiment words of opposite orientation tended not to co-occur at the sentence level. Popescu and Etzioni (2005) used a relaxation labeling method to find the semantic orientation of words in the context of given product features and sentences. They iteratively assigned semantic orientation labels to words by using various features including intra-sentential co-occurrence and synonyms of a thesaurus. Kanayama and Nasukawa (2006) used both inter-sentential and intra-sentential co-occurrence to get polarities of words and phrases.

However, most of the existed studies rely on only the relationship between words either in a thesaurus (Esuli & Sebastiani, 2005; Hu & Liu, 2004; Kamps et al., 2004; Kim & Hovy, 2004) or in a corpus (Gamon & Aue, 2005; Hatzivassiloglous & McKeown, 1997; Kanayama & Nasukawa, 2006; Popescu & Etzioni, 2005; Turney, 2002), while ignoring other relationships between words and documents (e.g., the relationship from words to documents, and the relationship from documents to words, the relationship between documents and documents), as well as the existed old-domain knowledge.

In order to uncover all these knowledge, in this paper, we design a novel algorithm to construct a domain sentiment lexicon. This algorithm is based on random walk model by taking into account all kinds of relationships among documents and words, from both old domain and new domain.

3. Proposed methods

3.1. Overview

A random walk model assumes that, from one period to the next, the original time series merely takes a random "step" away from its last recorded position (http://www.duke.edu/~rnau/411rand.htm). It has been applied to computer science, physics and a number of other fields.

In the field of information retrieval, given a graph whose edges are weighted by the probability of traversing from one node to another, one critical problem is how to rank the nodes. We can simulate a random walk on the graph. When a walker traverses from one node to another, he visits some notes more often than the others. So we can rank the nodes according to the probabilities that the walker will visit the nodes after the random walk. This idea has been successfully extended in many studies (e.g. PageRank (Brin, Page, Motwami, & Winograd, 1999), LexRank (Erkan & Radev, 2004), CollabRank (Wan & Xiao, 2008)). Our approach is also inspired by this idea.

We can get the following thoughts based on the ideas of PageRank and HITS (Kleinberg, 1998):

(1) If a word is strongly linked with other positive (negative) words, it tends to be positive (negative); and if a document is strongly linked with other positive (negative) documents, it tends to be positive (negative).
(2) If a word appears in many positive (negative) documents, it tends to be positive (negative); and if a document contains many positive (negative) words, it tends to be positive (negative).

Given the data points of words and documents, there are four kinds of relationships in our problem:

- WW-Relationship: It denotes the relationship between words, usually computed by knowledge-based approach or corpus-based approach.
- WD-Relationship: It denotes the relationship from words to documents, usually computed by the relative importance of a document to a word.

- DD-Relationship: It denotes the relationship between documents, usually computed by their content similarity.
- DW-Relationship: It denotes the relationship from documents to words, usually computed by the relative importance of a word in a document.

Fig. 1 gives an illustration of the relationships. In this figure, the upper plane denotes the new domain and the lower plane denotes the old domain. In the old-domain plane, the solid ellipses denote documents and the solid nodes in each ellipse denote the words within the document. Since the old-domain data are labeled, we use red and blue colors to denote the positive and negative sentiment respectively. In the new-domain plane, dotted ellipses denote documents and hollow nodes denote words.

The proposed approach could make full use of all the relationships in a unified framework. The framework consists of a graph-building phase and an iterative reinforcement phase. In the graph-building phase, the input includes both the labeled data from old domain and the unlabeled data from new domain. The proposed approach builds four graphs based on these data to reflect the above relationships. For old-domain data, we initialize every document and word a score ("1" denotes positive, and "−1" denotes negative) to represent its degree of sentiment orientation, and we call it sentiment score; for new-domain data, we set the initial sentiment scores to 0.

In the iterative reinforcement phase, our approach iteratively computes the sentiment scores of the words and documents based on the graphs. When the algorithm converges, all the sentiment words get their sentiment scores. If its sentiment score is between 0 and 1, the word should be classified as "positive". The closer its sentiment score is near 1, the higher the "positive" degree is. Otherwise, if its sentiment score is between 0 and −1, the word should be classified as "negative". The closer its sentiment score is near −1, the higher the "negative" degree is.

3.2. Sentiment-graph building

3.2.1. Symbol definition

In this paper, we have two document sets: the new-domain documents $D^U = \{d_1, \ldots, d_{nd}\}$ where d_i is the term vector of the ith text document and each $d_i \in D^U$ $(i = 1, \ldots, nd)$ is unlabeled; the old-domain documents $D^L = \{d_{nd+1}, \ldots, d_{nd+md}\}$ where d_j represents the term vector of the jth text document and each $d_j \in D^L$ $(j = nd+1, \ldots, nd+md)$ should have a label from a category set $C = \{\text{negative, positive}\}$. We assume the old-domain dataset D^L is from the related but different domain with the new-domain dataset D^U. Also, we have two sentiment word sets: $W^U = \{w_1, \ldots, w_{nw}\}$ is the word set of D^U and each $w_i \in W^U$ $(i = 1, \ldots, nw)$ is unlabeled; $W^L = \{w_{nw+1}, \ldots, w_{nw+mw}\}$ is the word set of D^L and each $w_j \in W^L$ $(j = nw+1, \ldots, nw+mw)$ has a label from C. Furthermore, we have two sentiment dictionaries: L^U is the sentiment dictionary of new domain and each word in L^U is unlabeled; L^L is the sentiment dictionary of old domain and each word in L^L has a label from C. Our objective is to maximize the accuracy of assigning a label in C to $w_i \in W^U$ $(i = 1, \ldots, nw)$ utilizing the old-domain data D^L and W^L in another domain.

The proposed algorithm is based on the following presumptions:

(1) $W^L \cap W^U \neq \Phi$.
(2) The labels of documents appear both in old domain and new domain should be the same.

In this section, we build four graphs to reflect four relationships, and the meanings of symbols are shown in Table 1. In this table, the first column denotes the name of the relationship; the second column denotes the similarity matrix to reflect the corresponding relationship; in consideration of convergence, we normalize the similarity matrix, and the normalized form is listed in the third column; in order to compute sentiment scores, we find the neighbors of a document or a word and the neighbor matrix is listed in the fourth column.

3.2.2. Document-and-word Graph

We build a weighted bipartite graph between documents and words in the following way: each node in the graph corresponds to a document in D^U and D^L or a word in W^U and W^L; if word w_j does not appear in document d_i, there is no edge between the two nodes; otherwise, we create an edge between d_i and w_j. The edge weight $wei(d_i, w_j)$ is proportional to the importance of word w_j in document d_i, computed as follows:

$$wei(d_i, w_j) = \frac{tf_{w_j} \times idf_{w_j}}{\sum_{w \in d_i} tf_w \times idf_w} \quad (1)$$

where w represents a unique word in d_i and tf_w, idf_w are respectively the term frequency in the document and the inverse document frequency.

In our approach, we need the relationship between D^U and W^U orW^L, and the relationship between W^U and D^U or D^L.

We use an adjacency matrix M to denote the similarity matrix between D^U and W^U orW^L, whose first nw columns denote the similarity matrix between D^U and W^U, and last mw columns denote the similarity matrix between D^U and W^L. Each entry of $M = [M_{ij}]_{ndx(nw+mw)}$ is corresponding to $wei(d_i, w_j)$.

In consideration of convergence, we normalize M to $\widehat{M}$ by making the sum of each row equal to 1:

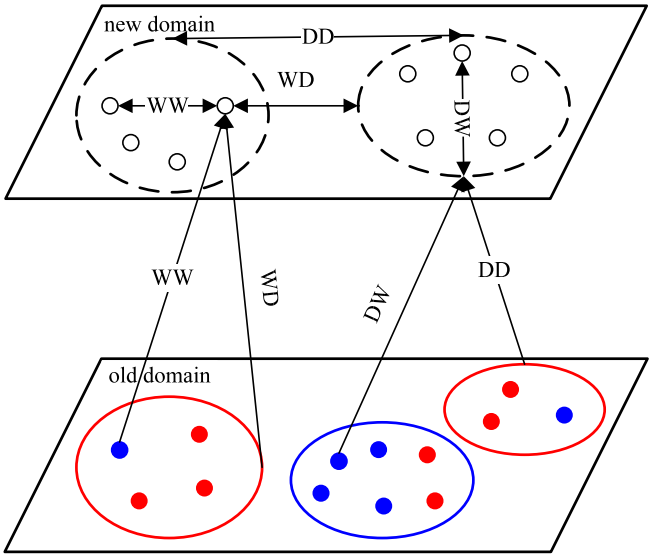

Fig. 1. Illustration of the four relationships.

Table 1
Symbol definition.

Relationship	Similarity matrix	Normalized form	Neighbor matrix
DW	$M = [M_{ij}]_{ndx(nw+mw)}$	$\widehat{M}$	$Mn = [Mn_{ij}]_{nd\times 2K}$
WD	$N = [N_{ij}]_{nwx(nd+md)}$	$\widehat{N}$	$Nn = [Nn_{ij}]_{nw\times 2K}$
DD	$U = [U_{ij}]_{ndx(nd+md)}$	$\widehat{U}$	$Un = [Un_{ij}]_{nd\times 2K}$
WW	$V = [V_{ij}]_{nwx(nw+mw)}$	$\widehat{V}$	$Vn = [Vn_{ij}]_{nw\times 2K}$

$$\widehat{M}_{ij} = \begin{cases} M_{ij} / \sum_{j=1}^{nw+mw} M_{ij}, & \text{if } \sum_{j=1}^{nw+mw} M_{ij} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In order to find the neighbors (in another word, the nearest documents or words) of a document from both W^U and W^L, we sort the similarity matrix between D^U and W^U and the similarity matrix between D^U and W^L in descending order separately, in order to get $\widetilde{M}$. That is, we firstly sort every row of $\widehat{M}_{ij}$ $(j = 1, \ldots, nw)$ in descending order and secondly sort every row of $\widehat{M}_{ij}$ $(j = nw + 1, \ldots, nw + mw)$ in descending order.

Then for $d_i \in D^U (i = 1, \ldots, nd)$, $\widetilde{M}_{ij} (j = 1, \ldots, K)$ corresponds to K neighbors in W^U, and $\widetilde{M}_{ij} (j = K + 1, \ldots, 2K)$ corresponds to K neighbors in W^L. We use a matrix $Mn = [Mn_{ij}]_{nd \times 2K}$ to denote the neighbors of D^U in W^U and W^L.

Similarly, we use an adjacency matrix $N = [N_{ij}]_{nwx(nd+md)}$ to denote the similarity matrix between W^U and D^U or D^L, whose first nd columns denote the similarity matrix between W^U and D^U, and last md columns denote the similarity matrix between W^U and D^L. Each entry N_{ij} is corresponding to $wei(d_j, w_i)$. We normalize N to $\widehat{N}$ to make the sum of each row equal to 1. Then we separately sort every row of $\widehat{N}_{ij} (j = 1, \ldots, nd)$ in descending order and every row of $\widehat{N}_{ij}$ $(j = nd + 1, \ldots, nd + md)$ in descending order, to get $\widetilde{N}$. Finally, we use a matrix $Nn = [Nn_{ij}]_{nw \times 2K}$ to denote the neighbors of W^U in D^U and D^L.

3.2.3. Document-and-document graph

We build an undirected graph whose nodes denote documents in both D^L and D^U and edges denote the content similarities between documents. If the content similarity between two documents is 0, there is no edge between the two nodes. Otherwise, there is an edge between the two nodes whose weight is the content similarity.

The content similarity between two documents is computed with the cosine measure. We use an adjacency matrix U to denote the similarity matrix, whose first nd columns denote the similarity matrix between D^U and D^U, and last md columns denote the similarity matrix between D^U and D^L. $U = [U_{ij}]_{ndx(nd+md)}$ is defined as follows:

$$U_{ij} = \begin{cases} \frac{d_i \cdot d_j}{\|d_i\| \times \|d_j\|}, & i \neq j, \\ 0, & i = j. \end{cases} \quad (3)$$

The weight associated with word w is computed with $tf_w\, idf_w$ where tf_w is the frequency of word w in the document and idf_w is the inverse document frequency of word w, i.e. $1 + \log(N/n_w)$, where N is the total number of documents and n_w is the number of documents containing word w in a data set.

We normalize U to $\widehat{U}$ to make the sum of each row equal to 1. Then we separately sort every row of $\widehat{U}_{ij}$ $(j = 1, \ldots, nd)$ in descending order and every row of $\widehat{U}_{ij} (j = nd + 1, \ldots, nd + md)$ in descending order, to get $\widetilde{U}$. Finally, we use a matrix $Un = [Un_{ij}]_{nd \times 2K}$ to denote the neighbors of D^U in D^U and D^L.

3.2.4. Word-and-word graph

Similar to the Document-and-Document Graph, we build an undirected graph to reflect the relationship between words in W^L and W^U, in which each node corresponds to a word and the edge weight between any different words corresponds to their semantic similarity.

We compute the semantic similarity using corpus-based approach which computes the similarity between words utilizing information from large corpora. There are many measures to identify word semantic similarity, such as mutual information (Kleinberg, 1998), latent semantic analysis (Turney, 2001) etc. In this study, we compute word semantic similarity based on the sliding window measure, that is, two words are semantically similar if they co-occur at least once within a window of maximum K_{win} words, where K_{win} is the window size. We use an adjacency matrix V to denote the similarity matrix, whose first nw columns denote the similarity matrix between W^U and W^U, and last mw columns denote the similarity matrix between W^U and W^L. $V = [V_{ij}]_{nwx(nw+mw)}$ is defined as follows:

$$V_{ij} = \begin{cases} \log \frac{N_w \times p(w_i, w_j)}{p(w_i) \times p(w_j)}, & \text{if } w_i \neq w_j \text{ and } i \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

where N_w is the total number of words in D^U; $p(w_i, w_j)$ is the probability of the co-occurrence of w_i and w_j within a window, i.e. $num(w_i, w_j)/N_w$, where $num(w_i, w_j)$ is the number of the times w_i and w_j co-occur within the window; $p(w_i)$ and $p(w_j)$ are the probabilities of the occurrences of w_i and w_j respectively, i.e. $num(w_i)/N_w$ and $num(w_j)/N_w$, where $num(w_i)$ and $num(w_j)$ are the numbers of the times w_i and w_j occur. We normalize V to $\widehat{V}$ to make the sum of each row equal to 1. Then we separately sort every row of $\widehat{V}_{ij}$ $(j = 1, \ldots, nw)$ in descending order and every row of $\widehat{V}_{ij}$ $(j = nw + 1, \ldots, nw + mw)$ in descending order, to get $\widetilde{V}$. Finally, we use a matrix $Vn = [Vn_{ij}]_{nw \times 2K}$ to denote the neighbors of W^U in W^U and W^L.

3.3. Proposed method

Based on the two thoughts introduced in Section 3.2, we fuse the four relationships abstracted from the three graphs together to iteratively reinforce sentiment scores, and we can obtain the iterative equation as follows:

$$ds_i = \gamma \times \sum_{g \in Un_{i\cdot}} (\widehat{U}_{ig} \times ds_g) + (1 - \gamma) \times \sum_{l \in Mn_{i\cdot}} (\widehat{M}_{il} \times ws_l) \quad (5)$$

$$ws_j = \gamma \times \sum_{g \in Nn_{j\cdot}} (\widehat{N}_{jg} \times ds_g) + (1 - \gamma) \times \sum_{l \in Vn_{j\cdot}} (\widehat{V}_{jl} \times ws_l) \quad (6)$$

where $i\cdot$ means the ith row of a matrix; $Ds = \{ds_1, \ldots, ds_{nd}, ds_{nd+1}, \ldots, ds_{nd+md}\}$ represents the sentiment scores of D^U and D^L; $Ws = \{ws_1, \ldots, ws_{nw}, ws_{nw+1}, \ldots, ws_{nw+mw}\}$ represents the sentiment scores of W^U and W^L; γ controls the relative contributions to the final sentiment scores from document sets and word sets.

The matrix form is

$$Ds = \gamma \times \widehat{U} Ds + (1 - \gamma) \times \widehat{M} Ws, \quad (7)$$

$$Ws = \gamma \times \widehat{N} Ds + (1 - \gamma) \times \widehat{V} Ws. \quad (8)$$

In consideration of the convergence, Ds and Ws are normalized separately after each iteration as follows to make the sum of positive scores equal to 1, and the sum of negative scores equal to -1:

$$ds_i = \begin{cases} ds_i / \sum_{j \in D^U_{neg}} (-ds_j), & \text{if } ds_i < 0, \\ ds_i / \sum_{j \in D^U_{pos}} ds_j, & \text{if } ds_i > 0, \end{cases} \quad (9)$$

$$ws_j = \begin{cases} ws_j / \sum_{i \in W^U_{neg}} (-ws_i), & \text{if } ws_j < 0, \\ ws_j / \sum_{i \in W^U_{pos}} ws_i, & \text{if } ws_j > 0, \end{cases} \quad (10)$$

where D^U_{neg} and D^U_{pos} denote the negative and positive document set of D^U respectively; W^U_{neg} and W^U_{pos} denote the negative and positive word set of W^U respectively.

Here is the complete algorithm:

1. Initialize the sentiment score *vector* ds_i of $d_i \in D^L$ $(i = nd + 1, \ldots, nd + md)$ with 1 when d_i is labeled "positive", and with -1 when d_i is labeled "negative", and initialize the sentiment score *vector*

ws_i of $w_i \in W^L$ $(i = nw+1,\ldots,nw+mw)$ with 1 when w_i is labeled "positive", and with -1 when w_i is labeled "negative". And we normalize ds_i $(i = \mathrm{nd}+1,\ldots,\ \mathrm{nd}+md)$ $(ws_i$ $(i = \mathrm{nw}+1,\ldots,\mathrm{nw}+mw))$ to make the sum of positive scores of $D^L(W^L)$ equal to 1, and the sum of negative scores of $D^L(W^L)$ equal to -1. Also, the initial sentiment scores of D^U and W^U are set to 0.

2. Alternate the following two steps until convergence:

2.1 Compute and normalize ds_i $(i = 1,\ldots,\ nd)$:

$$ds_i^{(k)} = \gamma \times \sum_{g \in Un_i} (\widehat{U}_{ig} \times ds_g^{(k-1)}) + (1-\gamma) \times \sum_{l \in Mn_i} (\widehat{M}_{il} \times ws_l^{(k-1)}),$$

$$ds_i^{(k)} = \begin{cases} ds_i^{(k)} / \sum_{j \in D^U_{neg}} (-ds_j^{(k)}), & \text{if } ds_i^{(k)} < 0, \\ ds_i^{(k)} / \sum_{j \in D^U_{pos}} ds_j^{(k)}, & \text{if } ds_i^{(k)} > 0. \end{cases}$$

2.2 Compute and normalize ws_j $(j = 1,\ldots,nw)$:

$$ws_j^{(k)} = \gamma \times \sum_{g \in Nn_j} (\widehat{N}_{jg} \times ds_g^{(k-1)}) + (1-\gamma) \times \sum_{l \in Vn_j} (\widehat{V}_{jl} \times ws_l^{(k-1)}),$$

$$ws_j^{(k)} = \begin{cases} ws_j^{(k)} / \sum_{i \in W^U_{neg}} (-ws_i^{(k)}), & \text{if } ws_j^{(k)} < 0, \\ ws_j^{(k)} / \sum_{i \in W^U_{pos}} ws_i^{(k)}, & \text{if } ws_j^{(k)} > 0, \end{cases}$$

where $ds_i^{(k)}$ and $ws_j^{(k)}$ denote the ds_i and ws_j at the kth iteration.

3. According to $ds_i \in Ds(i = 1,\ldots,nd)$, assign each $d_i \in D^U$ $(i = 1,\ldots,nd)$ a label. If ds_i falls in the range $[-1,0]$, assign d_i the label "negative"; if ds_i falls in the range $[0,1]$, assign d_i the label "positive".

4. Experiments

In this section, we evaluate our approach on three different domains and compare it with some state-of-the-art algorithms, and also evaluate the approach's sensitivity to its parameters.

4.1. Data preparation

We use three Chinese domain-specific data sets, which are: Electronics Reviews[1] (Elec, from http://detail.zol.com.cn/), Stock Reviews[2] (Stock, from http://blog.sohu.com/stock/) and Hotel Reviews[3] (Hotel, from http://www.ctrip.com/). All of these reviews are manually labeled as "negative" or "positive".

The detailed information of the data sets is shown in Table 2, which shows the name of the data set (Data Set), the number of negative reviews (Neg), the number of positive reviews (Pos), the average length of reviews (Length), the number of different words (Vocabulary) in this data set.

We use ICTCLAS (http://ictclas.org/), a Chinese text POS tool, to segment these Chinese reviews. Then, utilizing the part-of-speech tagging function provided by ICTCLAS, we take all adjectives, adverbs and adjective-noun phrases as candidate sentiment words. After removing the repeated words and ambiguous words, we get a list of words in each domain.

For the list of words in each domain, we manually label every word as "negative", "positive" or "neutral", and we take those "negative" and "positive" words as a sentiment word set.

[1] http://www.searchforum.org.cn/tansongbo/corpus/Elec-IV.rar.
[2] http://www.searchforum.org.cn/tansongbo/corpus/Sto-IV.rar.
[3] http://www.searchforum.org.cn/tansongbo/corpus/Htl-IV.rar.

Table 2
Data sets composition.

Data set	Neg	Pos	Length
Elec	554	1054	121
Stock	683	364	460
Hotel	2000	2000	181

Note that we use the sentiment word set only for source domain, while using the candidate sentiment words for target domain.

Lastly, the documents are represented by vector space model. In this model, each document is converted into bag-of-words presentation in the remaining term space. We compute term weight with the frequency of the term in the document.

We choose one of the three data sets as source-domain data D^L, and its corresponding sentiment word set as W^L; we choose another data set as target-domain data D^U, and its corresponding candidate sentiment words as W^U.

Table 3 presents the detailed information of sentiment word set of every data set. It shows the name of the data set that sentiment word set belongs to (Data Set), the number of different words (Vocabulary), the number of domain-independent negative words (INW), the number of domain-dependent negative words (DNW), the number of domain-independent positive words (IPW), the number of domain-dependent positive words (DPW).

We choose one of the three data sets as old-domain data D^L, and its corresponding word set as W^L; we choose another data set as new-domain data D^U, and its corresponding word set as W^U.

4.2. Baseline methods

In this paper we compare our approach with the following baseline methods which take into account only the relationship between words, ignoring the other relationships:

PMI: This is a corpus-based method to infer word semantic orientation (Turney & Littman, 2003). This method takes some labeled sentiment words as paradigm words to infer the semantic orientations of unlabelled words. In detail, we use the common part (sentiment words) of old-domain data and new-domain data as the paradigm words of the PMI method.

SM + SO: This is an improved PMI method, and we set up the experimental environment as the default configurations as Gamon and Aue (2005).

LE: This method applies a lexicon extension algorithm (Kanayama & Nasukawa, 2006). This method is an unsupervised method, so we take the common part (sentiment words) between old-domain data and new-domain data as the original lexicon, and set up the experimental environment as the default configurations as Kanayama and Nasukawa (2006).

4.3. Overall performance

In this section, we compare proposed approach with the three baseline methods. There are three parameters in our algorithm, K, K_{win}, and γ. We set K to 80, K_{win} to 10, and γ to 0.5 respectively.

Table 3
Sentiment word sets composition.

Data set	Vocabulary	INW	DNW	IPW	DPW
Elec	6200	242	90	298	124
Stock	13,012	567	112	343	89
Hotel	11,336	199	59	253	93

It is thought that the algorithm achieves the convergence when the changing between the sentiment score wu_i computed at two successive iterations for any $w_i \in W^U$ ($i = 1,\ldots,nw$) falls below a given threshold, and we set the threshold 0.00001 in this work. These parameters will be studied in parameters sensitivity section.

Tables 4 and 5 show the accuracy comparison between the three baselines and the proposed method on six tasks for domain-independent words and domain-dependent words.

As we can compare between the two tables, all approaches on domain-independent tasks outperform those on domain-dependent tasks. This indicates that constructing domain-oriented sentiment lexicon is more difficult than constructing domain-independent sentiment lexicon.

Also, Tables 4 and 5 show that PMI performs so poorly that it seems not in accord with the conclusion in Turney (2001). This is because PMI is affected by the corpus size very much. The experimental results provided by Turney (2001) are obtained by making use of search engine, and taking the whole Internet as the corpus. However, the corpus in our experiment is relatively small, which may bring much noise and make the co-occurrence information sparse. So this may lead to the poor performance of PMI. Nevertheless, the two tables show our approach performs well on relatively small-scaled corpus. Therefore, this result convinces us of the robustness of proposed algorithm.

Seen from Table 5, our algorithm produces much better performance than the baselines on almost all the six tasks. The greatest increase of accuracy is achieved by about 29% on the task "Elec-> Stock" compared to PMI. The average accuracy is also higher than all the baselines, and the greatest increase is achieved by about 11.5% compared to PMI. The great improvement compared with the baselines indicates that our approach performs very effectively.

4.4. Parameters sensitivity

In this section, we conduct experiments to show that our algorithm is not sensitive to these parameters.

When we evaluate the parameter K, we set K_{win} to 10, and γ to 0.5. And we change K from 10 to 100, an increase of 10 each. We experiment the sensitivity of K on six tasks as mentioned in Section 4.4. And the results are shown in Fig. 2. It shows that our algorithm

Table 4
Accuracy of domain-independent sentiment word classification.

	Baselines			Proposed method
	PMI	SM + SO	LE	
Elec → Stock	0.6971	0.6834	0.7132	**0.8996**
Elec → Hotel	0.7661	0.7752	0.8073	**0.8139**
Stock → Hotel	0.6791	0.7119	0.8181	**0.8214**
Stock → Elec	0.7054	0.7331	**0.8132**	0.7586
Hotel → Stock	0.8538	0.8799	0.8670	**0.8991**
Hotel → Elec	0.7412	0.7672	**0.8339**	0.7859
Average	0.7405	0.7585	0.8088	**0.8298**

Table 5
Accuracy of domain-dependent sentiment word classification.

	Baselines			Proposed method
	PMI	SM + SO	LE	
Elec → Stock	0.5783	0.6059	0.6311	**0.8686**
Elec → Hotel	0.6842	0.7349	0.7326	**0.7983**
Stock → Hotel	0.6875	0.7123	0.7358	**0.7821**
Stock → Elec	0.7064	0.7330	0.7343	**0.7526**
Hotel → Stock	0.7371	0.7642	0.7813	**0.8543**
Hotel → Elec	0.7208	0.7542	**0.7634**	0.7435
Average	0.6857	0.7174	0.7298	**0.7999**

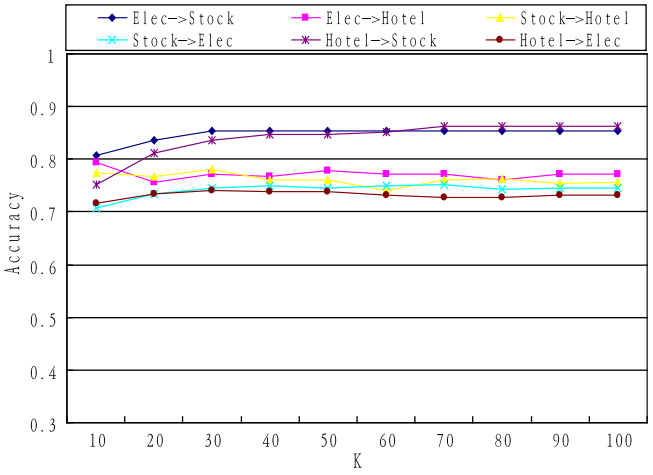

Fig. 2. Accuracy for different K.

is not very sensitive to K as long as K is large enough. We find the curves of these six examples rise gradually when K increases from 10 to 40, and the curves become stable after K arrives at 50. So we set K as 80 in our overall-performance experiment.

When we evaluate the parameter K_{win}, we set K to 80, and γ to 0.5. And we change K_{win} from 5 to 50, an increase of 5 each. We also experiment the sensitivity of K_{win} on six tasks as mentioned in Section 4.4. And the results are shown in Fig. 3. We find the curves of these six examples are almost stable when K_{win} increases from 5 to 50. It shows that our algorithm is not very sensitive to K_{win}, and so we set K_{win} to 10 in our overall-performance experiment.

To investigate the sensitivity of proposed method involved with the parameter γ, we set K to 80, and K_{win} to 10. And we change γ from 0 to 1, an increase of 0.1 each. We also evaluate γ on the six tasks mentioned above, and the results are shown in Fig. 5.

We can observe from Fig. 5 that the accuracy first increases and then decreases when γ is increased from 0 to 1. It is easy to explain this phenomenon. When γ is set to 0, this indicates our algorithm only uses word sets to aid classification, without the information of document sets. And if γ is set to 1, our algorithm only uses document sets to calculate sentiment score, without the help of word sets. Both cases above don't use all information of four relationships, so their accuracies are worse than to equal the contributions of both document and word sets. That is to say, both the words and the documents play an important role in the automatic construction of domain-oriented sentiment lexicon. Furthermore, this

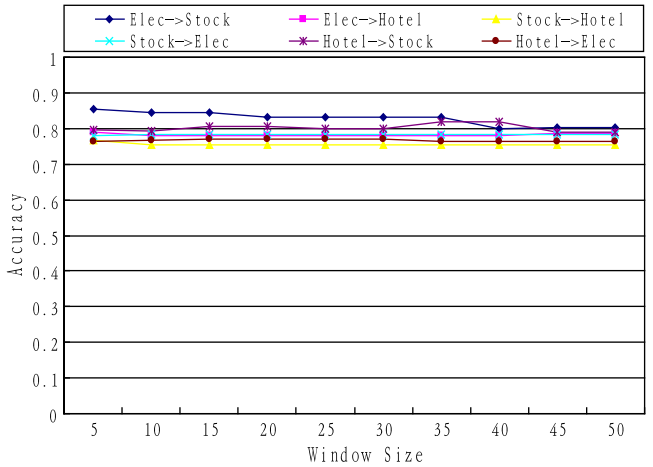

Fig. 3. Accuracy for different window size.

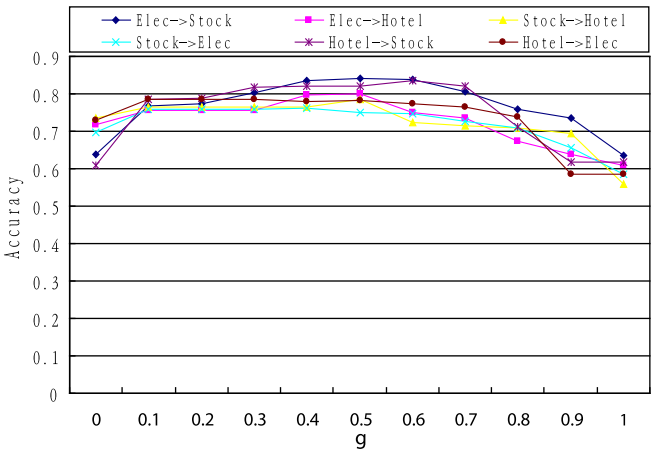

Fig. 4. Accuracy for different γ.

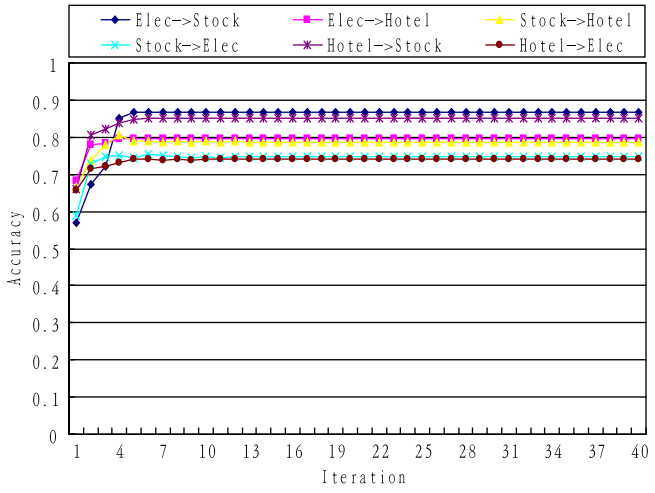

Fig. 5. Performance for iteration.

experiment shows that the proposed algorithm is not sensitive to the parameter γ when γ is from 0.4 to 0.7. We set γ to 0.5 in our overall-performance experiment.

4.5. Convergence

Our algorithm is an iterative process that will converge to a local optimum. We evaluate its convergence on the six tasks mentioned above. Fig. 6 shows the change of accuracy with respect to the number of iterations. We can observe from Fig. 6 that the curve rises sharply during the first 6 iterations, and it is very stable after 10 iterations are performed. This experiment indicates that our algorithm could converge very quickly to get a local optimum.

5. Conclusions

In this paper, we propose a novel approach to construct domain-oriented sentiment lexicon, which is based on random walk model by utilizing all the relationships among documents and words from both old domain and new domain.

We conduct experiments on three domain-specific sentiment data sets. The experimental results show that the proposed approach could dramatically improve the accuracy of identifying the polarities of sentiment words. In the same time, the experimental results indicate that no matter where the knowledge is from, i.e., the words or the documents, the old domain or the new domain, they all play an important role in the automatic construction of domain-oriented sentiment lexicon.

To investigate the parameter sensitivity, we conduct experiments on the same data sets. It is observed that our approach is not sensitive to its four parameters, and could converge very quickly to get a local optimum.

In this study, we employ only cosine measure, sliding window measure and vector measure to measure the relationships. In order to improve the performance of our approach, we will try other methods to calculate the similarity in the future. Furthermore, we experiment our approach on relatively small-scaled corpus, and we will apply our approach to bigger scaled corpus.

Acknowledgments

This work was mainly supported by two funds, i.e., 60933005 and 60803085.

References

Brin, S., Page, L., Motwami, R. & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web, Technical Report 1999-0120, Computer Science Department, Stanford University, Stanford, CA.

Du, W., Tan, S., Cheng, X. & Yun, X. (2010). Adapting information bottleneck method for automatic construction of domain-oriented sentiment Lexicon. In *Proceedings of WSDM*.

Erkan, G., & Radev, D. (2004). LexRank: graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research, 22*.

Esuli, A. & Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. In *Proceedings of CIKM*.

Gamon, M., & Aue, A. (2005). Automatic identification of sentiment vocabulary exploiting low association with known sentiment terms. In *Proceedings of ACL*.

Gamon, M. & Aue, A. (2005). Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *ACL-05 Workshop on Feature Engineering for Machine Learning in Natural Language Processing*.

Hatzivassiloglous, V. & McKeown, K. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of ACL*.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of KDD*.

Kamps, J., Marx, M., Mokken, R., & Rijke, M. (2004). Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC*.

Kanayama, H., & Nasukawa, T. (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of EMNLP*.

Kim, S., & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of COLING*.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. In *Procedings of ACM-SIAM symposium on discrete algorithms*.

Ku, L. W., Liang, Y. T., & Chen, H. H. (2006). Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-2006*.

Popescu, A. & Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of HLT/EMNLP*.

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting Naive Bayes to domain adaptation for sentiment analysis. In *Proceedings of ECIR*.

Tan, S., Wu, G., Tang, H. & Cheng, X. (2007). A novel scheme for domain-transfer problem in the context of sentiment analysis. In *Proceedings of CIKM*.

Tang, H., Tan, S., & Cheng, X. (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications, 36*(7), 10760–10773.

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for Chinese documents. *Expert Systems with Applications, 34*(4), 2622–2629.

Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of ECML 2001*.

Turney, P. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*.

Turney, P., & Littman, M. (2003). Measuring praise and criticism: inference of semantic orientation from association. *ACM Transactions on Information Systems, 21*(4), 315–346.

Wan, X., & Xiao, J. (2008). CollabRank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of Coling*.

Wu, Q., Tan, S., Zhai, H., Zhang, G., Duan, M., & Cheng, X. (2009). SentiRank: cross-domain graph ranking for sentiment classification. In *Proceedings of WI*.